Machine Learning for Probabilistic Modelling

Iain Murray

School of Informatics, University of Edinburgh
Hello! These slides were visual aids for a talk, and weren’t designed to be read. I’ve inserted some notes here to summarize what the points were supposed to be, and to give further references.

1. **Hierarchical modelling is essential.** Many models contain large numbers of ‘nuisance variables’. We need to learn how these are distributed, because if we make assumptions (including vague or so-called ‘uninformative’ ones), we’ll simply get wrong answers. The model I discussed was referring to: “Inferring the force law in the solar-system from a snapshot”, Bovy et al., 2010. 


Hierarchical models can be hard to infer. Example: [http://homepages.inf.ed.ac.uk/imurray2/pub/10hypers/](http://homepages.inf.ed.ac.uk/imurray2/pub/10hypers/)
2. Real-world models have a lot of messy detail:
1) complicated instrument-error distributions we may not care about;
2) theory encoded in expensive-to-run simulations.

Machine Learning can help.
If we don’t want wrong models (point 1.) we need to learn from large amounts of data about our instruments, and from simulation data describing our theories. I’ve been involved in a series of papers on flexible black-box probabilistic models that could be used here:
http://homepages.inf.ed.ac.uk/imurray2/pub/11nade/
http://homepages.inf.ed.ac.uk/imurray2/pub/13rnade/
http://homepages.inf.ed.ac.uk/imurray2/pub/14dnade/
4. Probabilistic inference methods need extending.
Approximate inference is a heavily-mined and active area. Getting up-to-speed and finding a niche is challenging. However, work in this area is important. In deep and wide graph structures, with billions of observations in some of the plates, it’s hard to do fully Bayesian inference.

Some of my work has been on identifying common small inference problems, which are usually only part of an analysis, and developing easier-to-use inference methods for them. E.g. http://homepages.inf.ed.ac.uk/imurray2/pub/10ess/

I’m now also interested in developing easier-to-use methods to summarize and communicate the results of local inferences across large models. I believe the way forward is fitting flexible representations of beliefs, by combining machine learning methods and approximate inference algorithms.
An Inference Problem
Acceleration law around the sun

\[ a(r) = -A \left( \frac{r}{r_0} \right)^{-\alpha} \]

From a snapshot:
8 planet positions and velocities
Graphical model

\[ \omega \rightarrow x^{(n)} \rightarrow e^{(n)} \]

\[ n=1..N \]
Hierarchical graphical model

\[ n=1..N \]

\[ \omega \]

\[ x^{(n)} \]

\[ e^{(n)} \]

\[ \theta \]
Inferences about the Sun
Priors on nuisance distributions
Priors on nuisance distributions
Gravitational exponent
Figure 1: Our unified graphical model (also known as a Bayesian network [27]), for astronomical image data. It integrates in a principled framework: large-scale cosmological models of galaxy and Milky Way formation; galaxy appearance models; spectral emission models and detailed camera, sky and telescope models. The shaded oval nodes are observed variables (i.e., their values are known) while the unshaded ones are unobserved and hence will be inferred from the raw astronomical data. The square nodes represent priors, typically informed by well-understood physics models. The arrows represent dependencies between variables in the model (and the lack thereof correspond to assumptions of independence). The conditional probability distributions within the model (which detail how a particular node depends on those variables which point to it) are not shown, but will be described in the text. The rectangles refer to replications of variables, e.g. an image will contain many stars/galaxies. The realization of this model is the ultimate goal of the project, but initial work will focus on sub-pieces of the model. This figure is best viewed in color.
Machine Learning?

— **Flexible probabilistic models**
  Neural networks and Gaussian processes

— **Inference methods**
  Statistical methods: MCMC, etc.
  Learning: recognition networks

Computational tools

Backpropagating derivatives is ubiquitous

Statistical computations need more support

\[ \theta \rightarrow \Sigma \rightarrow L \rightarrow F \]

\[ \frac{\partial \theta}{\partial F} \leftarrow \frac{\partial \Sigma}{\partial F} \leftarrow \frac{\partial L}{\partial F} \leftarrow F \]

The End

Reusable tools for solving inference problems
  Monte Carlo methods, Neural nets, and more

Derivative propagation for statistical computation